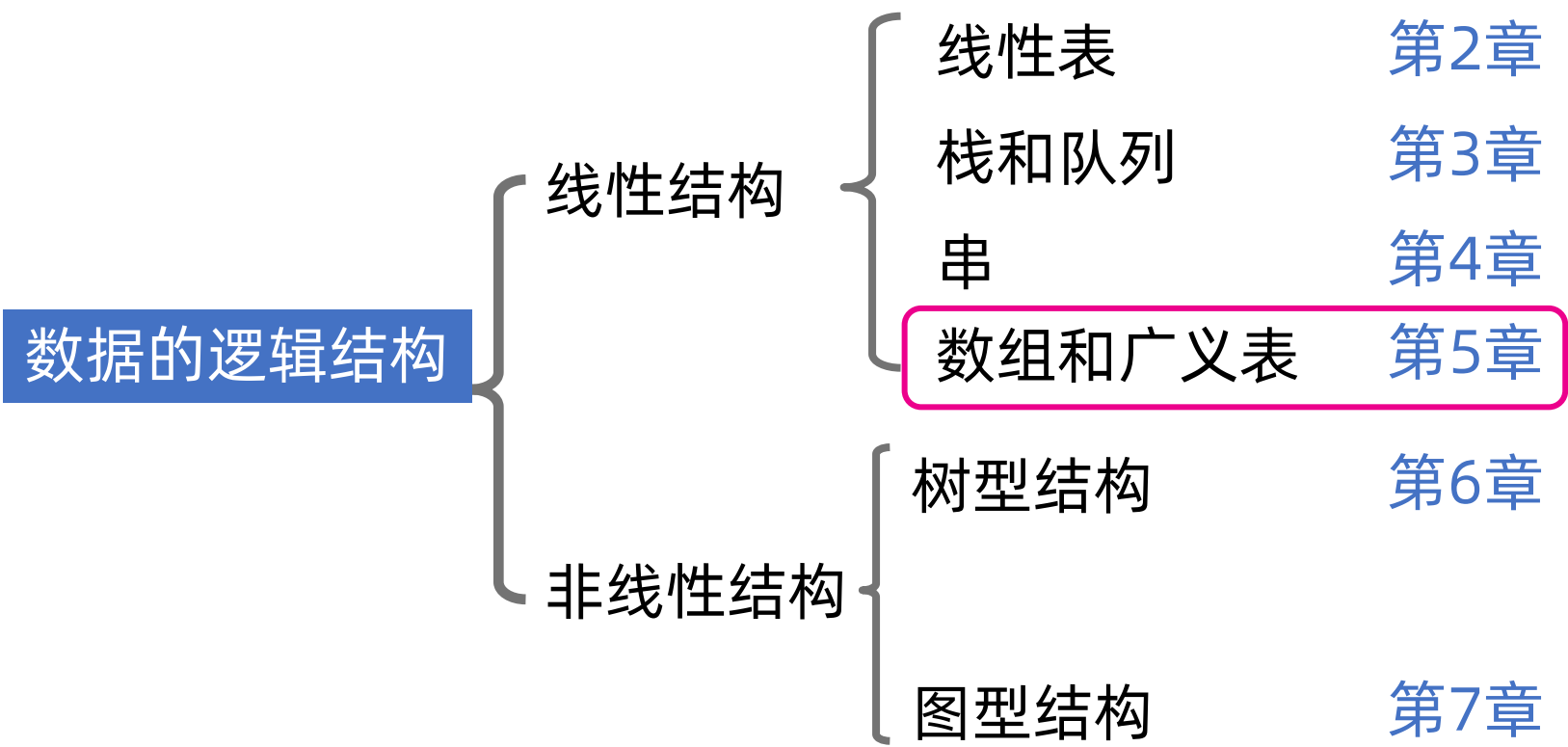


知识回顾



第5章 数组和广义表

DATA STRUCTURE

计算机科学学院 廖雪花

本章内容简介

数组和广义表

5.1 数组的定义

5.2 数组的顺序表示和实现

5.3 矩阵的压缩存储

5.4 广义表的定义

5.5 广义表的存储结构

5.1 数组的定义

廖雪花 LiaoXuehua

数组的基本概念

■ 数组的定义：

◆ 即数组是由 n 个具有相同数据类型的数据元素 a_1, a_2, \dots, a_n 组成的有限序列，且该有限序列必须存储在一块地址连续的存储单元中。

◆ 数组的下标：数组元素的位置。

- 一维数组
- 二维数组
- 多维数组

数组的基本概念

■注意：

(1)C语言的数组定义下标从0开始。

(2) 数组的处理相比其它复杂的结构要简单。

① 数组中各元素具有统一的类型；

② 数组元素的下标一般具有固定的上界和下界，即数组一旦被定义，它的维数和维界就不再改变。

③数组的基本操作比较简单。

数组的基本概念



■ 问题：数组与线性表的区别与联系？

✓ 相同之处：

它们都是若干个相同数据类型的数据元素 $a_0, a_1, a_2, \dots, a_{n-1}$ 构成的有限序列。

数组的基本概念

✓不同之处:

- (1)数组要求其元素占用一块地址连续的内存单元空间，而线性表无此要求；
- (2)线性表的元素是逻辑意义上不可再分的元素，而数组中的每个元素还可以是一个数组；
- (3)数组的操作主要是向某个下标的数组元素中存放数据和取某个下标的数组元素，这与线性表的插入和删除操作不同。

数组的定义

■ 二维数组

$$A = \begin{vmatrix} a_{00} & a_{01} & \dots & a_{0n-1} \\ a_{10} & a_{11} & \dots & a_{1n-1} \\ \dots & \dots & \dots & \dots \\ a_{m-1,0} & a_{m-1,1} & \dots & a_{m-1,n-1} \end{vmatrix}$$

□在此:

1. 可以将二维数组A看成是由m个行向量 $[X_0, X_1, \dots, X_{m-1}]^T$ 组成, 其中,
 $X_i = (a_{i0}, a_{i1}, \dots, a_{in-1})$, $0 \leq i \leq m-1$;
2. 可以将二维数组A看成是由n个列向量 $[Y_0, Y_1, \dots, Y_{n-1}]$ 组成, 其中,
 $Y_j = (a_{0j}, a_{1j}, \dots, a_{m-1j})^T$, $0 \leq j \leq n-1$ 。

数组的定义

■ 二维数组

$$A = \begin{vmatrix} a_{00} & a_{01} & \dots & a_{0n-1} \\ a_{10} & a_{11} & \dots & a_{1n-1} \\ \dots & \dots & \dots & \dots \\ a_{m-1,0} & a_{m-1,1} & \dots & a_{m-1,n-1} \end{vmatrix}$$

□ 由此可知：

二维数组中的每一个元素 最多可有二个直接前驱和两个直接后继（边界除外），故是一种**典型的非线性结构**。

二维数组的抽象数据类型定义

ADT Array2{

数据对象 : $D = \{a_{ij} | a_{ij} \in \text{ElemSet}, 0 \leq i \leq m - 1, 0 \leq j \leq n - 1\}$

数据关系: $S = \{R_1, R_2\}$

$R_1 = \{ \langle a_{i,j}, a_{i,j+1} \rangle | 0 \leq i \leq m - 1, 0 \leq j < n - 1 \}$

$R_2 = \{ \langle a_{i,j}, a_{i+1,j} \rangle | 0 \leq i < m - 1, 0 \leq j \leq n - 1 \}$

基本操作:

InitArray(&A, 2, m, n)

DestroyArray(&A)

value(A, &e, i, j)

Assign(&A, e, i, j)

} ADT Array2

数组的定义

■ 三维数组

- 三维数组最多可有三个直接前驱和三个直接后继，三维以上数组可以作类似分析。
- 因此，可以把三维以上的数组称为多维数组，多维数组可有多多个直接前驱和多个直接后继，故多维数组是一种非线性结构。

数组的定义

■ n维数组的抽象数据类型定义

ADT Array_n{

数据对象:

$$D = \{a_{j_1 j_2 \dots j_n} \mid a_{j_1 j_2 \dots j_n} \in \text{ElemSet}, \text{其中 } j_i = 0, 1, \dots, b_{i-1}, i = 1, \dots, n\}$$

数据关系:

$$S = \{R_1, R_2, \dots, R_n\}$$

$$R_i = \{ \langle \underline{a_{j_1 \dots j_i \dots j_n}}, a_{j_1 \dots j_{i+1} \dots j_n} \rangle \mid \begin{array}{l} 0 \leq j_k \leq b_{k-1}, 1 \leq k \leq n \text{ 且 } k \neq i, \\ 0 \leq j_i \leq b_{i-2}, \\ a_{j_1 \dots j_i \dots j_n}, a_{j_1 \dots j_{i+1} \dots j_n} \in D \end{array} \}$$

数组的定义

■ n维数组的抽象数据类型定义

基本操作:

InitArray(&A,n,bound1,..., boundn)

DestroyArray(&A)

value(A,&e,index1,...,indexn)

Assign(&A,e,index1,...,indexn)

} ADT Arrayn

本节要点

■ 数组的定义：

- ✓ 一维、二维、三维、n维
- ✓ 数组的抽象数据类型定义

■ 数组的特点

- ✓ 数组中的数据元素具有相同的数据类型。
- ✓ 数组中的数据元素个数是固定的。
- ✓ 数组一旦被定义，它的维数和维界就不再改变。因此，除了结构的初始化和销毁之外，数组只有存取元素和修改元素值的操作。

感谢聆听